REMARKS

Claims 1-8, 10-19, 21-26 and 28-30 are pending in the present application. Claims 4, 10, 12, 15, 21, 23 and 26 are amended. Amendments were made to the claims to provide strict antecedent basis for the terms used. These amendments do not change the scope of the claims. Claims 9, 20, and 27 are canceled. Claims 28, 29, and 30 are added. The features in these claims are supported in the specification at least on page 13, lines 6-15. Consequently no new matter has been added. Reconsideration of the claims is respectfully requested.

Applicant thanks Examiner Peyton for all the courtesies extended during the November 17, 2005 telephone interview. During the interview, Applicant's representative discussed the prior art of record and the manner in which the references, *Bahl* and *Groath*, fail to teach or disclose the features recited in the presently claimed invention in the independent claims. The arguments discussed as well as additional reasons that the claims are not anticipated are set forth in the remarks below. Examiner Peyton indicated an error in the Office Action dated September 21, 2005. On page 2, paragraph 3 of the Office Action, claims 4-27 are to read claims 15-27.

I. 35 U.S.C. § 102, Anticipation, Claims 15-27

The examiner rejects claims 15-27 under 35 U.S.C. § 102(e) as being anticipated by Bahl *et al.* (U.S. Patent No. 6,834,341 B1) ("*Bahl*"). This rejection is respectfully traversed.

As to claims 15-27, the examiner states:

> As per claims 15-27, *Bahl* teaches a method apparatus for detecting errors comprising means for setting a time span for a time window (event window); means for starting the time window; means responsive to the time window ending, for determining whether one or more input/output errors occur on a device path during a time window; and means responsive to one or more input/output errors occurring on the device path during the time window, for incrementing an error count. (Abstract, cols. 19, lines 42-col. 20, lines 1-58, Figs. 10-12)
>
> *Bahl* teaches a method of dynamic compression in a data path performed by a PANS server or a client machine. *Bahl* teaches of defining an event window which monitors the data path wherein a time frame for the event window is based

> on the specific type of monitoring define by the system. *Bahl's* event window monitors for errors that occur during the event window's time frame. A min/max error count is established and is tracked in order to determine if a specific number of errors occur more than a predetermined number of times (Fig. 11).

Office Action dated September 21, 2005, pages 2-3.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed Cir. 1994). *Bahl* does not teach every element of the claimed invention arranged as they are in the claims.

Currently amended claim 15 reads as follows:

> 15. An apparatus for detecting errors in a device path, the apparatus comprising:
>
> means for setting a time span for a time window based on a time to process a successful input/output command;
>
> means for starting the time window;
>
> means, responsive to the time window ending, for determining whether one or more input/output errors occur on a device path during a time window;
>
> means, responsive to one or more input/output errors occurring on the device path during the time window, for incrementing an error count;
>
> means for determining whether the error count reaches a predetermined limit; and
>
> means, responsive to the error count reaching the predetermined limit, for disabling the device path.

Bahl does not teach each and every feature of claim 15. More specifically, *Bahl* does not teach the feature of setting a time span for a time window based on a time to process a successful input/output command, and *Bahl* does not teach the feature of disabling the device path if the error count reaches a predetermined limit.

With respect to the features of setting a time span for a time window based on a time to process a successful input/output characteristics, and disabling the device path if

the error count reaches a predetermined limit, the examiner references **Figures 10 – 12** in *Bahl*. Each of these figures will be discussed in turn. **Figure 10** is shown as follows:

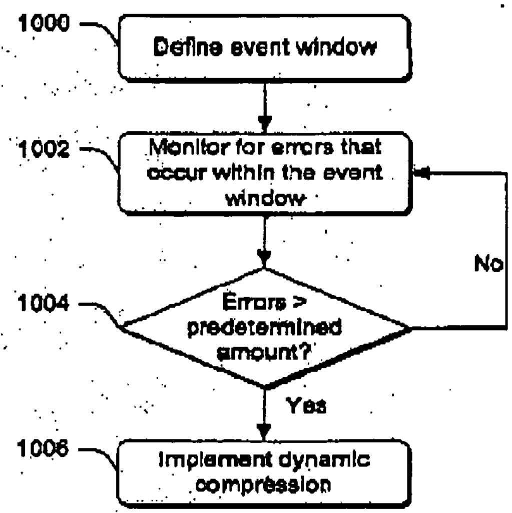

Fig. 10

Bahl, **Figure 10.** As can be seen in this figure, step **1000**, define event window, is nonspecific. There is no teaching indicating a feature of setting a time span for a time window based on a time to process a successful input/output. Further, step **1006** teaches implementing dynamic compression, rather than disabling a device path, as a consequence to exceeding the predetermined error count. The examiner cites the description of this figure which reads as follows:

> FIG. **10** is a flow diagram that describes steps in a dynamic compression method in accordance with the described embodiment. Dynamic compression can be performed by both the PANS server and the client machine. Step **1000** defines an event window within which monitoring takes place. The event window can be any suitable time frame for which monitoring is desired. Step **1002** monitors for errors that occur within the event window. The errors that can be monitored for include, without limitation bit errors, packet errors and the like. Monitoring can take place using any suitable monitoring techniques as will be understood

> by those of skill in the art. Accordingly, monitoring techniques are not discussed in detail any further.

Bahl column 19, lines 42-54. Although step **1000** is referred to as defining an event window, this section of *Bahl* describes step **1000** as any suitable time frame for which monitoring is desired. *Bahl's* **Figure 10** does not teach or disclose setting a time span for a time window based on a time to process a successful input/output command as recited in claim 15. *Bahl* does not address specifically setting the time span in the event window on any parameter. As the cited reference states, monitoring techniques are not discussed in any detail. In other words, no disclosure of an element in which setting a time span for a time window based on a time to process a successful input/output command is present in this section of *Bahl*, or any section of *Bahl*.

Bahl in **Figure 10** does not teach disabling a device path as a consequence to exceeding an error limit. Instead, *Bahl* teaches implementing dynamic compression as a response to exceeding an error limit.

Further description of **Figure 10** included in *Bahl* is as follows:

> Step **1004** determines whether the errors that occur are greater than a predetermined amount. Errors can be accounted for in any suitable way. For example, the gross number of errors that occur in a given time period can be determined. Alternately, the error rate can be determined. During this time, a base line compression can be employed by the client and the PANS server. A base line compression can comprise using a certain compression algorithm or variation thereof. In addition, a base line compression can comprise compressing a certain amount of the data packets (e.g. a certain percentage) within the event window. If the errors exceed the predetermined amount, then step **1006** implements dynamic compression.

Bahl, column 19, lines 54-66. As cited in the further description of **Figure 10**, *Bahl* does not teach disabling a device path as a consequence of exceeding an error limit as claim 15 recites, but rather, step **1006** implements dynamic compression as the consequence of exceeding an error count.

Turning now to the second figure cited by the examiner, **Figure 11** does not teach the features of setting a time span for a time window based on a time to process a successful input/output command, and disabling the device path if the error count reaches a predetermined limit. **Figure 11** is shown as follows:

1100

Errors	Compression percent
0-1	100%
2-5	80%
6-10	50%
11-15	10%
+15	0%

Fig. 11

Bahl, **Figure 11**. **Figure 11** is a look up table that relates the number of errors to the percentage of data compression that should be implemented. **Figure 11** does not teach setting a time span for a time window, nor does it teach disabling a device path. The description for **Figure 11** is as follows:

> **FIG. 11** shows a look up table generally at **1100** that can be used, in one embodiment, to implement dynamic compression. Here, the look up table **1100** contains two fields--an error field and a compression percent field. In this example, there are 5 entries in the error field, i.e. 0-1, 2-5, 6-10, 11-15, +15. These entries constitute different thresholds for errors that can occur within the event window. Each of the entries in the error field is associated with a compression percent. In this example, the compression percentages range from 100% for when there are very few detected errors, to 0% for when there are a large number of detected errors. Accordingly, as the data packets are transmitted, as long as the detected errors in an event window do not rise above 1, all of the data packets in the event window will be compressed. If, for example, the detected errors rise to between 6-10, then the percentage of data packets that get compressed

drops to 50%. This helps to ensure that during periods of transmission disruption, less of the data that is transmitted between the PANS server and the client are compressed thereby reducing the amount of data that is ultimately corrupted.

Bahl, column 29, lines 13-33. The description of **Figure 11** describes the relationship between the thresholds for error that can occur within the event window and the recommended percentage of compression. The description of **Figure 11** does not teach setting a time span for a time window, nor does it teach disabling a device path as a response to the error count reaching the predetermined limit, as claim 15 recites.

Turning now to the third figure cited by the examiner, **Figure 12** is shown as follows:

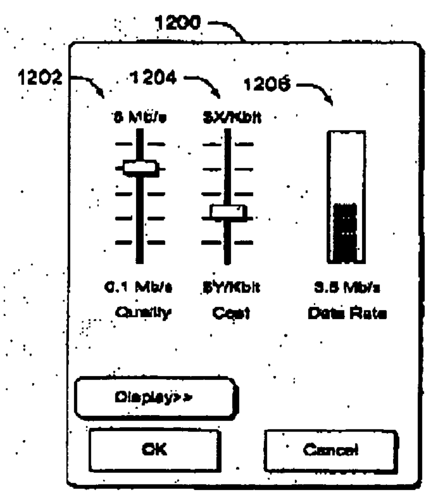

Fig. 12

Bahl, **Figure 12. Figure 12** shows an exemplary graphic user interface generally at **1200.** In this example, interface **1200** includes a bandwidth selector **1202** that is configured to enable a user of the computing device to adjust the bandwidth that is allocated to the computing device.

The text describing **Figure 12** reads as follows:

> **Fig. 12** shows an exemplary graphic user interface generally at **1200.** Interface **1200** is configured for display on a client computing

> device. In this example, the interface **1200** includes a bandwidth selector **1202** that is configured to enable a user of the computing device to adjust the bandwidth that is allocated to the computing device. Accordingly, a user is given the choice of the bandwidth allocation that they can receive. Interface **1200** also includes a cost selector **1204** that is configured to enable a user of the computing device to adjust the cost that is associated with the bandwidth that is allocated to the computing device. In this example, each of the selectors **1202, 1204** are sliders that can be manipulated with a user input device such as a mouse. By adjusting the cost (or the bandwidth allocation), the user can adjust the allocated bandwidth that they use to transmit their data packets. Accordingly, if a user is in a hurry (e.g. between flights in a busy airport), they could simply adjust one or both of the selectors to automatically select a high level of service that is available. In addition, a data rate display **1206** is provided that displays indicia of a data rate that is currently being provided to the computing device. This gives the user real time feedback so that they can confirm that they are in fact receiving the level of service that they selected and for which they will be charged.

Bahl, column 20, lines 35-58. As seen in this figure and in the cited paragraph above, the quality of the customer's service can be selected through the selection bar on this screen. *Bahl's* **Figure 12** teaches selecting a quality of service, not a means for setting a time span for a time window based on a time to process a successful input/output command or a means, responsive to the error count reaching the predetermined limit, for disabling the device path, as recited in claim 15. This figure does not provide any teaching to determine a time span, as recited in claim 15. The reference through this figure only discloses a customer selection option that indicates the ramifications of the choice the customer has made. *Bahl* does not teach a means for setting a time span or disabling a device path, as recited in claim 15. Thus, *Bahl* teaches customer selection of quality, not setting a time span or disabling a device path.

Finally, the examiner references the *Bahl* abstract. The abstract reads as follows:

> Systems and methods for providing network access, e.g. Internet access, are described. An architecture includes a host organization network through which network access is provided. The host organization network can be advantageously deployed in public areas such as airports and shopping malls. An authentication/negotiation component is provided for authenticating various users and negotiating for services with service providers on behalf of the system users. The authentication/negotiation component can include one or more specialized servers and a policy

> manager that contains policies that govern user access to the Internet. An authentication database is provided and authenticates various users of the system. An access module is provided through which individual client computing devices can access the Internet. In one embodiment, the access module comprises individual wireless access points that permit the client computing devices to wirelessly communicate data packets that are intended for the Internet. In one aspect, users are given a variety of choices of different service levels that they can use for accessing the Internet. The service levels can vary in such things as bandwidth allocation and security measures. The various service levels can be purchased by the users using their computing devices.

Bahl, abstract. The *Bahl* abstract teaches a method or a service wherein customers can connect to the Internet in a public place by using the *Bahl* invention. The customers will be able to choose the level of service from the provider by selecting a bandwidth. The customers will then be able to monitor the quality of the service they receive. *Bahl* does not teach a means for setting a time span for a time window, nor does *Bahl* teach a means for disabling the device path.

In summary, no mention of setting a time span for a time window based on a time to process a successful input/output command is present in this cited reference. Therefore, *Bahl* does not teach the feature of setting the time span for a time window element of claim 15 in the reference as believed by the examiner. *Bahl* does not teach setting the time span for a time window (presumably, the examiner interprets an event window to be a time window) based on a time to process a successful input/output command. Neither **Figures 11** nor **12** address setting the time span for a time window based on a time to process a successful input/output command. **Figure 11** shows a look up table generally at **1100** that can be used to implement dynamic compression. **Figure 12** shows an exemplary graphic user interface generally at **1200**. The examiner states, that *Bahl* teaches defining an event window which monitors the data path wherein a time frame for the event window is based on the specific type of monitoring defined by the system. These figures in *Bahl*, however, do not teach setting a time span for a time window based on a time to process a successful input/output command.

Since setting the time span for a time window based on a time to process a successful input/output command feature is explicitly recited in independent claims 15 and 26, setting the time span for a time window based on a time to process a successful

input/output command must be addressed in any rejection of the corresponding claims. Further, since disabling a device path based on the monitored errors is explicitly recited in amended independent claims 15 and 26, disabling the device path must be addressed in any rejection of the corresponding claims.

Since claims 16-19, 21-25, 29 and 30 depend from claims 15 and 26, the same distinctions between *Bahl* and the invention recited in claim 15 are valid for these claims. Additionally, the dependent claims recite other additional combinations of features not suggested by the reference.

For example, claim 16 reads as follows:.

> 16. The apparatus of claim 15, further comprising:
> means, responsive to an input/output command returning successful, for determining a new time span for the time window.

Claim 16 recites determining a new time span for the time window in response to a successful input/output command. This feature is not shown in *Bahl* as believed by the examiner. The examiner does not point with any specificity to any passage of the reference to support this feature. However, *Bahl* is directed towards providing Internet access and the user/customer's ability to monitor the service that is provided. Nowhere in *Bahl* is the feature of determining a new time span for the time window, responsive to an input/output command returning successfully, taught.

Further other features of the dependent claims are not taught by the reference. For example, claim 19 reads as follows:

> 19. The apparatus of claim 18, further comprising:
> means for setting a start time of the new time window equal to the timestamp of the input/output command.

Claim 19 recites a means for setting a start time of the new time window. This feature is not shown in *Bahl*. *Bahl* teaches Internet access not monitoring input/output events or setting a start time of the new time window. Again the examiner does not point out with any specificity this feature, as the examiner believes is taught, in *Bahl*.

Furthermore, *Bahl* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *Bahl* actually teaches away

from the presently claimed invention because it teaches a nonspecific event window with the time span undefined, as opposed to setting a time span for a time window based on a time to process a successful input/output command, as in the presently claimed invention. Absent the examiner pointing out some teaching or incentive to implement *Bahl* and the setting of the time span, one of ordinary skill in the art would not be led to modify *Bahl* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Bahl* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

II. 35 U.S.C. § 102, Anticipation, Claims 1-27

The examiner rejects 1-27 under 35 U.S.C. § 102(e) as being anticipated by Groath *et al.* (U.S. Patent No. 6,571,285 B1) ("*Groath*"). This rejection is respectfully traversed.

As to claims 1-27, the examiner states:

> *Groath* teaches a method for detecting errors in a network path comprising: setting a time span for a time window based on a time to processor [sic] a successful input/output network command; and monitoring a network device (col. 10, lines 4-6) input/output streams during the time window to detect sequential and distributed errors (col. 2, lines 7-23, col. 10, lines 2-8, 15-32, and col. 11, lines 43-62) having a sensitivity that is user definable. *Groath* further teaches method apparatus for detecting errors comprising means for setting a time span for a time window (monitoring window); means for starting the time window; means responsive to the time window ending, for determining whether one or more input/output errors occur on a device path during a time window; and means responsive to one or more input/output errors occurring on the device path during the time window, for incrementing an error count. (*Groath*, Abstract, col. 117, lines 45-col. 118, lines 1-44)

Office Action dated September 21, 2005, pages 3-4.

Claim 1 reads as follows:

1. A method for detecting errors in a device path, the method comprising:
setting a time span for a time window based on a time to process a successful input/output command; and
monitoring a device input/output stream during the time window to detect sequential and distributed errors having a sensitivity that is user definable.

Groath does not teach each and every feature of this claim. More specifically, *Groath* does not teach the feature of setting a time span for a time window based on a time to process a successful input/output command, and *Groath* does not teach the feature of monitoring a device input/output stream during the time window to detect sequential and distributed errors having a sensitivity that is user definable.

With respect to the feature of setting a time span for a time window based on a time to process a successful input/output network command, the examiner cites the following section of *Groath*:

> One embodiment of the present invention is composed of multiple software programs which are linked together to create an architecture which is capable of monitoring a network for events and checking system functions and resources. Such events can include alarms, faults, alerts etc. Other embodiments of the present invention may each include an individual software program.

Groath, column 10, lines 2-8. As can be seen, *Groath* teaches that software programs may be linked together to create the capability of monitoring a network. Linking software programs together to create the capability of monitoring a network is not the same as setting a time span for a time window based on a time to process a successful input/output command. Nowhere in this cited section does *Groath* teach setting a time span for a time window, as recited in claim 1.

Further the examiner states, "...input/output stream during the time window to detect sequential and distributed errors (col. 2, lines 7-23, col. 10, lines 2-8, 15-32, and col. 11, lines 43-62) having a sensitivity that is user definable" asserting that *Groath* teaches, detecting sequential and distributed errors. The first paragraph cited by the examiner reads as follows:

A method providing service assurance for a network to maintain an agreed upon Quality of Service. First, an alarm is generated to indicate a status of a network. The generation of the alarm comprises selecting a parameter of network to be monitored; determining a triggering level of the parameter; monitoring the parameter of an occurrence of the triggering level; and initiating alarm notification upon the monitored occurrence of the triggered level. Network event information is then dispatched upon generation of the alarm and is subsequently mapped. The data collected on the status of the network into a master file; reformatting the data into a standardized format; translating the data to key codes; sorting the data according to predetermined criteria; and concatenating the sorted data together. The data is then stored in a database. Thereafter, network availability is conveyed graphically.

Groath, column 2, lines 7-23. In this cited passage, *Groath* is teaching a monitoring scheme that does not include setting a time span for a time window. Rather the monitoring scheme taught in the cited passage of *Groath* teaches initiating an alarm notification based upon the occurrence of the triggering level. *Groath* has no teaching of setting a time span or time window in which to detect errors, as does claim 1 in the present invention. Neither does the passage, indicated by the examiner, teach detecting sequential or distributed errors.

The examiner also cited column 10, lines 2-8, these lines of *Groath* were cited and discussed above. As can be seen above, this cited section, teaches the potential to link software programs to implement monitoring. The cited section also teaches another embodiment which may provide for monitoring in a single software program. However, these portions of *Groath* do not teach setting a time span for a time window based on a time to process a successful input/output command, nor does *Groath* teach monitoring a device input/output stream during the time window to detect sequential and distributed errors having a sensitivity that is user definable.

Next, the examiner cited the section of *Groath* as follows:

Data Acquisition

In another embodiment of the present invention, Data Acquisition scripts are programs which coordinate the collection and transfer of application logs to a central location. Data Acquisition can be used so that log files containing performance statistics gathered by a monitoring program can be transferred to a central server for processing by the Performance Data Manipulator (see below). The Data Acquisition scripts

> may be written in PERL.
> Performance Data Manipulator (PDM)
>
> In an embodiment of the present invention, a PDM is a script that processes log files that have been collected by Data Acquisition in order to load the data into a database. The PDM converts the log files from formats specific to a particular monitoring program into a common format. PDM then formats the file based on data warehousing techniques which include converting nodes and performance metrics to key codes which are stored in the database. The coded data file is then bulk loaded into the database. The PDM may be written in PERL.

Groath, col. 11, lines 43-62. This section of *Groath* teaches data acquisition scripts and examples for processing and storing performance statistics of interest in *Groath*. These three sections cited by the examiner do not point out the features recited by claim 1. There are no teachings in *Groath* concerning a method for setting a time span for a time window based on a time to process a successful input/output command. *Groath*, however, teaches monitoring general quality indices on a network, storing and reporting the indices. Indeed there are no teachings in *Groath* specifically addressing input/output command timing. The examiner does not specifically point out where in *Groath* this feature is taught. Again, that while verbiage teaches a general monitoring system with detail to methods of data storage, the features of claim 1 are absent from this passage. *Groath* does not teach setting a time span for a time window based on a time to process a successful input/output command, nor does it teach monitoring a device input/output stream during the time window to detect sequential and distributed errors having a sensitivity that is user definable.

The examiner further states:

> *Groath* further teaches method apparatus for detecting errors comprising means for setting a time span for a time window (monitoring window); means for starting the time window; means responsive to the time window ending, for determining whether one or more input/output errors occur on a device path during a time window; and means responsive to one or more input/output errors occurring on the device path during the time window, for incrementing an error count. (*Groath*, Abstract, col. 117, lines 45-col. 118, lines 1-44)

Office Action, dated September 21, 2005, pages 3-4. The abstract is as follows:

> A method providing service assurance for a network to maintain an agreed upon Quality of Service. First, an alarm is generated to indicate a status of a network. The generation of the alarm comprises selecting a parameter of network to be monitored; determining a triggering level of the parameter; monitoring the parameter of an occurrence of the triggering level; and initiating alarm notification upon monitored occurrence of the triggering level. Network event information is then dispatched upon generation of the alarm and is subsequently mapped. The data collected on the status of the network is then manipulated by concatenating the data collected on a network into a master file; reformatting the data into a standardized format; translating the data to key codes; sorting the data according to predetermined criteria; and concatenating the sorted data together. The data is then sorted in a database. Thereafter, network availability is conveyed graphically.

Groath, abstract. *Groath* teaches selecting a network to be monitored, determining a triggering level of the parameter, monitoring the parameter of an occurrence of the triggering level, and initiating an alarm notification upon monitored occurrence of the triggering level. Further, *Groath* teaches how the data collected is stored and communicated. The cited section above does not teach setting a time span for a time window based on a time to process a successful input/output command, as does claim 1. Nor does the cited section teach monitoring a device input/output stream during the time window to detect sequential and distributed errors having a sensitivity that is user definable. Setting a time span or a time window is not taught in the *Groath* abstract, or indeed anywhere in the *Groath* reference. Nothing in the abstract indicates detecting sequential and distributed errors.

Further, the examiner cited the following section of *Groath*:

> Event Handling Requirements
>
> FIG. 31 illustrates an embodiment of the present invention which maps events on a network with service assurance capabilities. In operation **3100**, a network is monitored for the occurrence of availability events, threshold events, and trap events. At least one occurred event is correlated to at least one other occurred event in operation **3102** to generate at least one correlating event. In operation **3104**, the occurred events and correlating events are mapped on at least one network map. The network map is subsequently displayed in operation **3106**.
>
> In one embodiment of the present invention, the step of monitoring the network further comprises: tracking the availability of individual components of network for events, tracking the availability of individual

services of the network for events, tracking the availability of individual processes of an operating system of the network for events, tracking the status of agent processes on individual components of the network for events, monitoring the operating system and application performance of network for threshold events, and monitoring traps of the network for events.

In yet another embodiment, the network map is a node level map and/or an event level map. The node level map displays node responding events, agent not responding events, and/or node down events. The step of mapping the occurred events and correlating events when the network map comprises the event level map further comprise additional steps. In particular, the occurred events and correlating events may be filtered based upon predetermined criteria. The filtered events may also be mapped on the event level map. In still yet another embodiment, at least one notification action is generated based upon the occurred events and/or correlating events.

Availability Events

This section details the three availability events monitored. Node Up and Node Down/Interface Up and Interface Down-Tracking an individual network component (such as a router, server, workstation, etc.). This will be tracked using Network Node Manager and ECM. We will track all nodes which we monitor on the test network, including the following: nsmmws16, nsmmws09, twmmnt02, twmmdb02, nsmmrt03, nsmmrt04. When a Node or Interface fails to respond to a ping, a Node or Interface Down event for the specific node will be generated. When a Node or Interface responds to a ping after immediately after failing to respond to a ping, a Node or Interface Up for the specific node event will be generated.

Service Up and Service Down-Tracking an individual network service (FTP, NNTP, POP3, SMTP, DNS, HTTP, and RADIUS (RADIUS is tracked as a probe only)). This will be tracked using the Collector Internet Service Monitor. When a service fails to respond to the ISM a Service Down event for the specific service will be generated. When a Service responds to the ISM immediately after failing to respond to the ISM, a Service Up event for the specific service will be generated.

Process Up and Process Down-Tracking an individual process on a Windows NT or UNIX system. This will be tracked using Patroller Process KMs. When the number of specific processes with a specific name running under a specific owner changes from a preset number, a Process Down event will be generated for the specific process, owner and number of processes. When the number of specific processes with a specific name running under a specific owner changes back to the preset number from another number, a Process Up event event will be generated for the specific process, owner, and number of processes.

Groath, column 117, line 45 – column 118, line 44. The first paragraph of the cited section above teaches correlating error events to other events to produce a correlated error. This is not the same as setting a time span of a time window or monitoring a device input/output stream during the time window to detect sequential and distributed errors having a sensitivity that is user definable. The cited section of *Groath* then continues giving some level of detail and example for two embodiments of the *Groath* invention. *Groath* teaches monitoring a network; however, the reference does not teach specifically monitoring an input/output stream, nor does *Groath* teach differentiating between sequential and distributed errors. Instead the reference teaches monitoring Nodes and agents in a network. The reference does not teach how to set the time span on a time window, nor does it teach detecting sequential and distributed errors. This section does teach indicating status of the Node as up if immediately following an error event or ping, a successful event ping follows. This is not the same as detecting sequential and distributed errors. It would be impermissible hindsight to project the monitoring of sequential and distributed errors on a network input/output stream into the *Groath* reference. Consequently, the rejection of claim 1 has been overcome.

Independent claims 4, 15, and 26 contain features similar to claim 1. Additionally, the features in amended claims 4, 15, and 26 are not taught in *Groath*. Amended claim 4 is as follows:

> 4. A method for detecting errors in a device path, the method comprising:
> setting a time span for a time window based on a time to process a successful input/output command;
> starting the time window;
> responsive to the time window ending, determining whether at least one input/output error occurs on a device path during the time window;
> responsive to one or more input/output errors occurring on the device path during the time window, incrementing an error count by one;
> determining whether the error count reaches a predetermined limit; and
> responsive to the error count reaching the predetermined limit, disabling the device path.

Groath does not teach the response of disabling the device path when the error count reaches the predetermined limit, instead *Groath* teaches, as seen in the sections cited above, methods of storing, correlating and reporting monitored data. Therefore, rejections of claims 4, 15, and 26 are overcome.

Additionally, claims 2-3, 5-8, 10-14, 16-19, 21-25 and 28-30 recite other additional combinations of features not suggested by the reference, such as the method in claim 2, which reads as follows:

> 2. The method of claim 1, wherein the time span is dependent upon system loads and variations in transport speeds.

The examiner points to no specific section of the *Groath* teaching this feature. Nowhere in *Groath* are these features taught. Additionally, the dependent claims recite other additional combinations of features not suggested by the reference.

Claim 16 reads as follows:

> 16. The apparatus of claim 15, further comprising:
> means, responsive to an input/output command returning successful, for determining a new time span for the time window.

Claim 16 recites determining a new time span for the time window in response to a successful input/output command. This feature is not shown in *Groath* as believed by the examiner. The examiner does not point with any specificity to any passage of the reference to support this feature.

Furthermore, *Groath* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *Groath* actually teaches away from the presently claimed invention because it teaches setting a Node to up if a successful ping is returned immediately after an error event takes place on that Node. This is different than detecting distributed errors which can take into account error events occurring non-sequentially as in the presently claimed invention. *Groath* teaches ways of storing, retrieving and displaying data as opposed to setting a time span for a time window based on a time to process a successful input/output command as in the presently

claimed invention. Absent the examiner pointing out some teaching or incentive to implement *Groath* and setting a time span based on a time to process an input/output comment, one of ordinary skill in the art would not be led to modify *Groath* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Groath* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

III. Conclusion

It is respectfully urged that the subject application is patentable over *Bahl* and *Groath* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: 12/14/05

Duke W. Yee

Duke W. Yee
Reg. No. 34,285
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

DWY/mam